December 2005

Dimensions of Partial Completion of Activities in Workflow Management

Georg Peters
Munich University of Applied Sciences

Roger Tagg
University of South Australia

Dimensions of Partial Completion of Activities in Workflow Management

Georg Peters
Munich University of Applied Sciences
Department of Computer Science/Mathematics
80637 Munich, Germany
Email: georg.peters@muas.de

Roger Tagg
University of South Australia
School of Computer and Information Science
Mawson Lakes, SA 5095, Australia
email: roger.tagg@unisa.edu.au

Abstract

Process and workflow management have become well accepted methods for helping to increase efficiency in companies. However the experience has been that highly structured workflows need to be more flexible to meet the requirements of dynamic businesses today. Recently Lin and Orlowska [2005] have addressed this topic by introducing partially complete-able activities. In contrast to classic workflows the activities do not show an all-or-nothing behaviour (they are fully completed or not) but can be completed on different levels. The decision as to whether or not the process can be continued is then the responsibility of the process owner. In our paper we extend this concept by the introduction of fuzzy and probably complete-able activities. Moreover we show that a process memory can further increase the flexibility of the workflow.

Keywords

Workflow management, process modelling, partial completion of activities, flexibility

INTRODUCTION

Process modelling and workflow management have gained increasing importance for companies wanting to improve their business processes and better manage their performance [van der Aalst, van Hee 2002]. This is both true within a specific organisation, and – with the increasing prevalence of business-to-business (B2B) e-commerce – between organisations as well [Marinescu 2002, Riempp 1998 and the Crossflow project (www.crossflow.org)].

However, limits to the effective adoption of workflow management have been encountered, and attributed to a weakness, in many commercial software packages, in support for flexibility. A range of dimensions of flexibility was discussed by Tagg [2003]. One example is the ease by which a workflow schema or pattern can be changed to reflect a change in the underlying business process.

Another example is the ease with which an individual workflow instance (or business case) can be allowed to diverge from the general pattern. This is typically required because processes fall behind schedule and need to be got back on track by such means as increasing resources or taking agreed short cuts. One specific type of short cut is to allow progression of the workflow before some activities have been fully completed. But in virtually all commercial workflow management systems (WfMS), an activity is only considered as completed when all its post-conditions have been fulfilled.

To increase flexibility in workflow systems Lin and Orlowska [2005] have suggested relaxing this constraint by allowing partial completion of activities. This means that only a certain number of the standard post-conditions must be met to proceed with the process. The decision of whether the process can be continued - or the level of completion of activity must be further increased - has to be taken by the stakeholders of the activity and the process as a whole.

Allowing partial completion of activities can lead to a higher flexibility of the process and a better alignment to real life situations. According to Lin and Orlowska [2005] the main advantages obtained by this concept are:

- a reduced processing time
- earlier release of resources for other activities.

In our paper we extend the concept of partially completed activities by distinguishing two dimensions, the *fuzzy* completion and the *probability* of the completion of an activity. The first refers to the partial completed activity in the sense of Lin and Orlowska. Here we show that fuzzy set theory can be utilized to formulate this concept. In the second case we assume a certain probability that the activity is fully completed.

Furthermore we introduce a memory dimension. In a memory-free process, the degree of completion of an earlier activity cannot influence the succeeding activities. In a process with a memory, allowing partial completion of an activity early in a process can be compensated by requiring stricter conditions on the completion of an activity in a later step of the process.

The paper is organized as follows. In the next section we give a short introduction to flexible workflow management and describe the concept of partial completion of activities according to Lin and Orlowska [2005]. In section three we extend the model of Lin and Orlowska by the introduction of a process memory, then we introduce the probability and fuzzy dimensions of partial completion. The paper concludes with a summary.

PARTIAL COMPLETION OF ACTIVITIES

Flexibility in Workflow Management

Workflow management, as instanced in a number of commercial software tools, involves the definition of one or more workflows. Each workflow is a dependency graph, or set of rules, which constrain the order in which work should be done in each of a number of instances (business cases) that follow the same workflow pattern. Workflow can be applied to automated tasks (by invoking the programs that perform the tasks), or manual tasks (by alerting a human user through a task list or by email).

Workflow differs from traditional groupware, where most activity is performed as and when the human participants choose. In general, workflow process paths and completion criteria for activities are pre-defined; while in groupware, both are decided ad hoc (Figure 1). This makes groupware more flexible and workflow systems more structured.

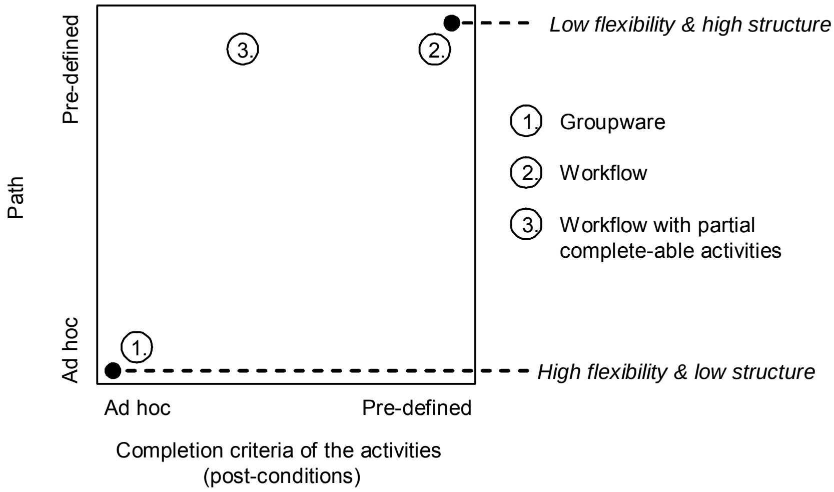

Figure 1: Flexibility dimensions of workflow and groupware

Examples of WfMSs that cater well for flexibility are mostly still in research laboratories rather than commercial products. Examples are Adept [Reichert 1998] and Chameleon [Sadiq 2000]. As well as allowing for migration of instances from a current workflow schema to a new one. Chameleon also allows instance modification.

Partial Completion of Activities

In workflow systems the completion of an activity is defined by the fulfilment of *all* of its post-conditions. As long as the post-conditions are not fully completed the workflow system does not proceed to the next activity. This behaviour can be characterized as *all-or-nothing* strategy and therefore shows a rather inflexible behaviour Lin and Orlowska [2005]. Critical path based project management systems class this as *finish-to-start* dependency, but usually the only alternative is *start-to-start*, which does not model reality well.

To achieve more flexibility in activity-to-activity dependencies, Lin and Orlowska [2005] introduced *partial complete-able activities* with the following properties:

- The objective of the activity is decomposable: the activity can be completed on different levels denoted as L_1, …, L_N, where L_N indicates a full completion of the activity (note: this appears to imply an order in a single dimension).
- For each activity, minimum requirements (denoted by the level L_M) are defined which must be met in any case to secure required basic quality standards.
- Absolute completion of the activity is not critical for the process: a partially completed activity (at least on level L_M) does not directly endanger the overall outcome of the process.
- The progress of the activity must be monitored at a fine grain to determine the level of its completion.

The decision process whether an activity is completed or not goes as follows (see Figure 2):

- As long as the minimum requirements are not met the workflow systems does not proceed to the next activity. We refer these activities as *insufficiently completed activities*.
- If the minimum requirements are met and a process stakeholder (in most cases the process owner, however in decentralized processes the performer of the activity could suffice) considers the activity as sufficiently completed, then the workflow system proceeds to the next activity. We refer these activities as *sufficiently completed activities*.
- If the full completion level L_N is reached the workflow system automatically continues with the next activity. We refer these activities as *fully completed activities*.

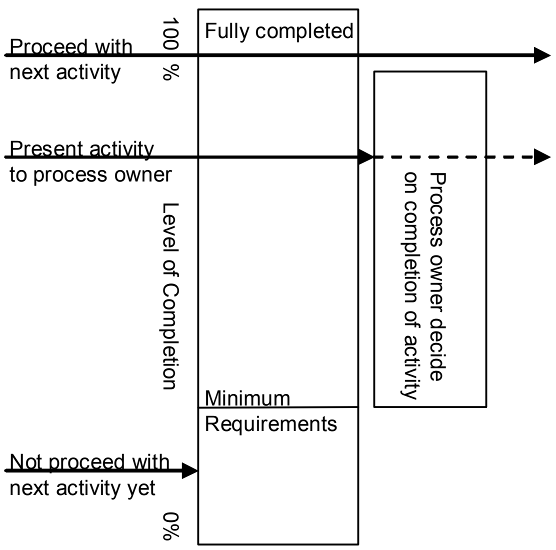

Figure 2: Fulfilment of Requirements Levels

Partial complete-able activities increase the flexibility of workflow systems (③ in Figure 1). An alternative solution would be to decompose the objectives and the activities into sub-objectives (elementary objective) and their corresponding sub-activities (elementary activities). This would make the situation amenable to a classic workflow system with a clearly defined relationship between the activity and its completion, and would bypass the need for any concept of partial completion.

However, Lin and Orlowska [2005] give the following reasons to support their concept of partial complete-able activities instead of breaking-up the given activity into elementary activities:

- Splitting of the activity into sub-activities would increase transaction costs and possibly system overheads also.
- Splitting also requires that for the aggregated activity (a) a single resource is responsible for the completion of all the activity and that (b) a single performer must fulfil the objective at the same time.

To formally model the concept of partial completion, Lin and Orlowska [2005] developed an enhanced activity finite state machine as show in Figure 3. The model is a generalization of the state model of the Workflow Management Coalition [1995].

It goes as follows: An activity can have the states *open* and *closed*. In the open state it is *scheduled* and therefore not active at first: *open.not_active.scheduled*. When the performer picks the activity out of a work-item list it changes its state to *open.active.commenced*. If a sub-objective of the activity is completed the state changes to *open.active.partly_completed*.

As already defined above the level of completion is indicated by L_1, L_2, …, L_M, …, L_N with L_N the level of full achievement of the objective of the activity and L_M the minimum requirements. If the activity reaches the levels L_M, … L_{N-1} the process owner can decide that the activity is sufficiently completed. In that case the activity changes to its final state *closed.completed*. This state will also be reached when the level L_N is achieved since the workflow engine automatically closes the activity.

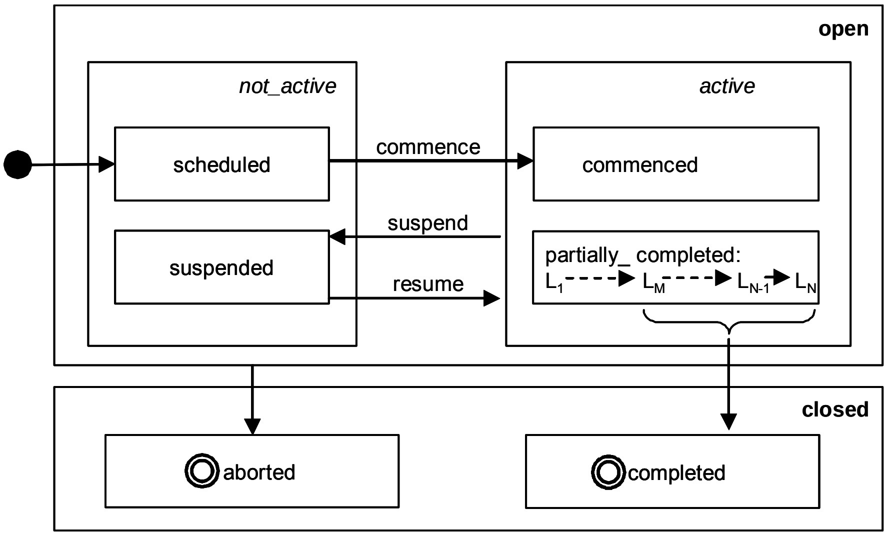

Figure 3: Enhanced Activity Finite State Machine (according to Lin and Orlowska [2005])

Anytime during the activity is *open.active.commenced* it can be set on “Wait”: *not_active.suspended* and *resumed* accordingly. The activity also can be aborted at any time: *closed.aborted*.

DIMENSIONS OF PARTIAL COMPLETED ACTIVITIES

Memory and Memory-Free Processes

The model of Lin and Orlowska [2005] is memory-free: once the process owner regards a partially completed activity as completed the activity reaches the state *closed.completed*. The process continues and has no memory; in particular the level of completion is not recorded. Whether the activity is only partially or fully completed results in the same state *closed.completed* and has no influence on any further process step.

However, the assumption of memory-freeness gives the process a reduced flexibility in comparison to a process with memory. In such a generalized model the degree of completion has an impact on future decision spaces within the process.

For example take the process “Prepare an excellent three course meal” consisting of the activities “Prepare Starter”, “Prepare Main Dish” and “Prepare Dessert” (see Figure 4).

Figure 4: “Prepare an Excellent Three Course Meal” Process

In a *memory-free* process every step of the process must meet minimum requirements independently from the other process steps. Once the minimum requirements of an activity are met the chef is free to define it as completed and continue with the next activity. The overall objective of the process can only be taken into account indirectly since no trade-off between the degree of completion of the single activities is possible.

In contrast, in a *process with memory* the overall objective is considered as well. In our example an excellent three course meal could have one weaker course at most. This means the starter *xor* (exclusive or) the main dish *xor* the dessert

may not fulfil the maximum requirements. If the starter and the main dish are perfect then the meal can still be classified as excellent even if the dessert is a little bit too sweet.

Now trade-offs between the activities of the process can be formulated. A perfect starter and a great main dish compensate for a slightly weak dessert. Obviously this gives the process a higher degree of flexibility in comparison to the memory-free process. This seems a more appropriate model for real business scenarios, since B2B partners will probably only accept partial completion of an early activity if the rest of the objective is going to get fixed later.

The reasons for not formulating the process as one activity and then applying the concepts of partially completed activities are as follows:

- An activity would become very complex if its scope covered everything that could be compensated. In our example the preparation of the three course meal would melt down to only one, very complex activity.
- A basic rule of thumb for design processes is that a single resource is responsible for one activity [van der Aalst, van Hee 2002]. However this rule would compete with the trade-off rule as defined above. The one activity "Prepare a three course meal" would be assigned to at least two resources: the chef responsible for the starter and the main dish and a confisier for the dessert.

A process with memory has a distinct advantage in comparison to the memory-free approach as proposed by Lin and Orlowska [2005], as the minimum objective of the overall process can be more flexibly met. In the case of a memory-free process one always has to assume the worst case scenario. This is when all activities just reach their minimum requirements. Obviously these minimum requirements must meet higher standards in comparison to a process with memory where trade-offs between high and low performing activities are possible. When only one activity is completed on a higher level then the process objective is also accomplished at a higher degree than desired. Generally this leads to a waste of resources in a memory-free process.

However a process with memory has three significant disadvantages in comparison with the memory-free process:

- It can only be applied when trade-offs between the goals of the activities are possible.
- Designing and running the process is more complex than designing and running a memory-free process. In the design phase all the trade-offs must be defined. In the running phase the workflow system must additionally monitor and record the degrees of completion of each activity and adapt the required degree of completion of further process steps.
- The possible trade-off between low and high accomplishment of activities might encourage performers of early activities to only meet the minimum requirements, which tends to result in stricter requirements in later process steps.

These have to be taken into account when an organisation has to decide on introduction of a process memory.

The Fuzzy Dimension

Fuzzy Sets as Indicators for Completion

Fuzzy set theory was introduced by Zadeh in 1965. Fuzzy sets can be interpreted as a measurement for neighbourhood relations: how similar is the actual output of an activity to a given post-conditions. The indicator for the similarity in fuzzy set theory is called membership degree $\mu=[0,\ldots,1]$. A membership degree $\mu=1$ indicates that the output fully equals the required post-conditions while a membership degree $\mu=0$ shows a total dissimilarity between output and post-conditions.

This method is very similar to the requirements as suggested by Lin and Orlowska. Therefore fuzzy set theory can be applied to the concept of partially completed activities. The main advantage is that one can use this well established theory with its formal tools to formulate the partially completeness of the activities.

For example we now can apply fuzzy concepts like aggregation and decision making [e.g. Zimmermann 2001] to this problem. Especially the aggregation concepts are useful when one formulates trade-offs between the degrees of completion of the activities.

In our example, let us assume that a perfect dessert of our three course meal requires exactly 0.3 ounce of sugar: $\mu=1.0$. No sugar and more than 0.6 ounce of sugar would lead to totally unacceptable results indicated by membership degrees of $\mu=0.0$ (see Figure 5).

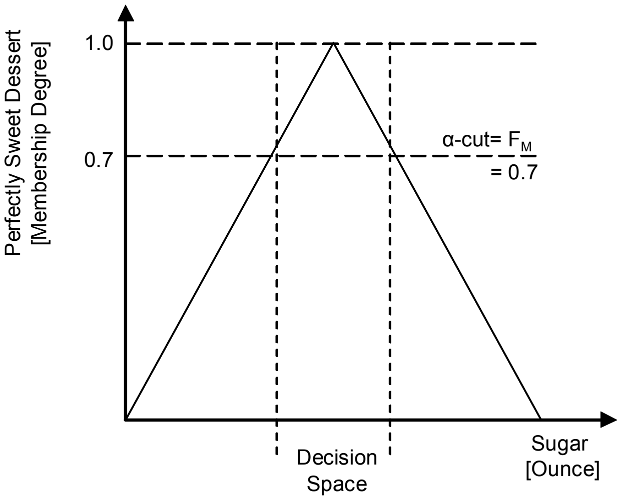

Figure 5: Fuzzy Decision Sugar

Acceptable results are defined by a dessert that has at least a membership degree (similarity) of $\mu=0.7$ to the post-condition "perfectly sweet dessert". Since one cuts off any solution below the required minimum membership degrees F_M as insufficient this is referred as *α-cut* in fuzzy terminology.

In Figure 5 the mapping of the membership degrees on the x-axis shows the ounces of sugar that lead to the decision space in which the process owner is free to decide whether the dessert is completed or not. Therefore in our example any weight between 0.21 and 0.91 ounce is acceptable to proceed with the process.

Sufficiently and Fully Completed Activities

To easily manage the workflow system we suggest a more detailed state model by the introduction of the states *closed.completed (F_i)* where Fi ($F_M \leq F_i \leq 1$) indicates the level of completion of the activity.

This equals the membership degree as describe above. An activity with the *state closed.completed (0.8)* indicates that it belongs to the set completed with a membership degree of $\mu=0.8$.

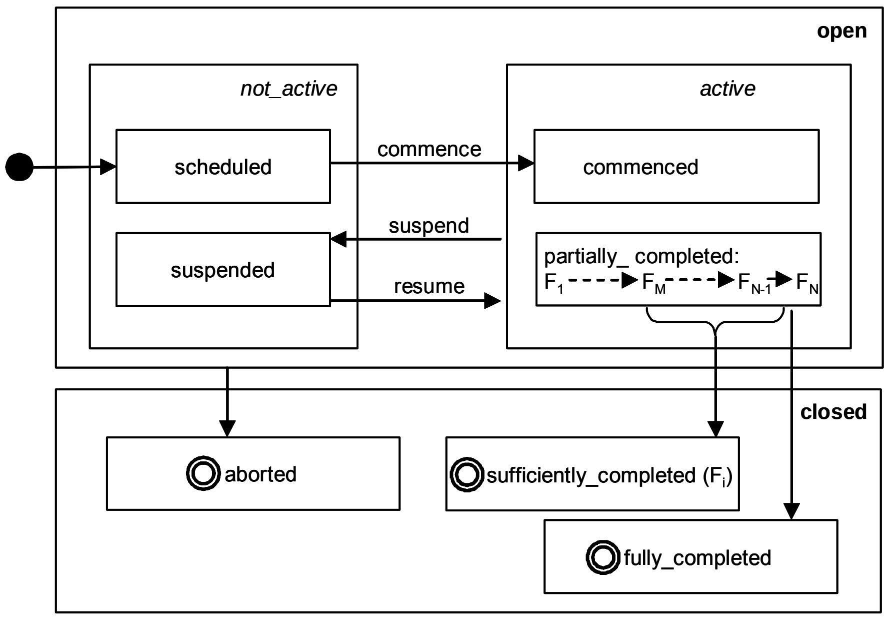

Figure 6: Fuzzy Enhanced Activity Finite State Machine

For operations departments and process owners it also makes sense to explicitly distinguish between fully and only partially completed activities. This leads to the following abbreviations:

- state: *closed.completed (F_i)* with $F_M \leq F_i < 1.0$: *sufficiently_completed (F_i)*

- state: *closed.completed (F_i)* with F_i=1.0: *fully_completed*

The fuzzy enhanced activity finite state machine is show in Figure 6. Our extension of Lin and Orlowska [2005] model now clearly separates between the levels of completion and therefore has a finer granularity.

Note, that in general the discrete levels $F_1, \ldots, F_N$ are replace by a (continuous) membership function as shown in Figure 5. This membership degree can be derived out of one indicator, in our example the sugar in ounces, or an aggregation of membership degrees of several indicators.

Let us assume that a good dessert is not only defined by the portion of sugar but by its "look" also. For simplicity we take the colour (wavelength) of the dessert as indicator: the more red the dessert looks the better (Figure 7).

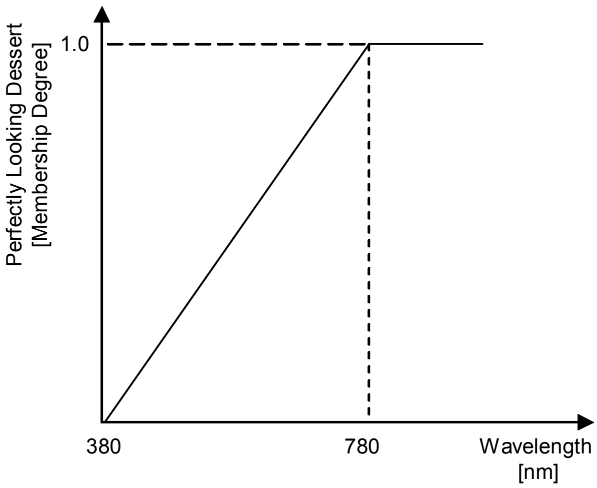

Figure 7: Fuzzy Decision Colour

To obtain the membership degree of the combined sugar&colour decision the single membership degrees sugar (e.g. μ_S=0.8) and colour (e.g. μ_C=0.5) are be summed up by an fuzzy aggregation operator, for example the basic min-operator: $\mu_{aggregated}=\min\{\mu_S,\mu_C\}=\min\{0.8, 0.5\}=0.5$. Note that the min operator has no compensatory power. For more advanced operators with compensation properties see e.g. Dubois, Prade [1982], Werners [1988], Yager [1980] or Zimmermann, Zysno [1980].

The Probability Dimension

Probability as Indicators for Completion

Besides using the similarity measure as shown in the previous section, the process owner could also decide to proceed with the process when sufficient evidence is given that a certain status is reached by the activity. Here we deal with the "insecurity" that is the uncertainty over whether or not the outcome actually matches the intended post-conditions of the activity.

For example, take the following claim process of a car insurance company (see Figure 8): although only the original invoice of a repair of a car garage gives incontrovertible proof of the claimed insurance sum (= post-condition) the insurer also accepts a facsimile of the receipt (actual outcome) as first proof.

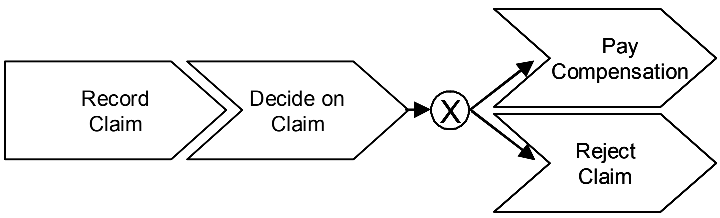

Figure 8: "Claim" Process

Generally the insurer now has two different policies:

- Policy 1. The facsimile of the receipt is accepted as sufficient proof although there is a certain probability that the facsimile is faked. A double-check against the original receipt would be more expensive than the limited number of frauds. Therefore the process continues memory-free: the claim is accepted under no further reserve. The workflow system does not require any roll-back strategies.

- Policy 2. The fax is accepted as first proof that leads to a pay-out of the insurance sum. However the pay-out is done under reserve that the original receipt will be received in a given time period. Here the workflow system needs roll-back strategies in case that the original is not received in time. This policy needs an advanced transaction management [e.g. Leymann, Roller 2000].

Along the lines of the discussion on fuzzy trade-offs we introduce a compensation between outcomes that surely meet the requirements of the activity and outcomes that only probably meet these requirements.

For example, consider a placement process to fill a job vacancy. The overall objective of the process is to hire the employee who fits best to the task description of the job. The selection process consists of two activities: the evaluation of the application letter and a personal interview with the short listed candidates. Obviously both steps lead only to probabilities (denoted as P_{AL} for the application letter and P_{PI} for the personal interview) that the candidate really meets the expectations of the company.

In a memory-free process both activities must meet minimum requirements, say a confidence level of $P_{AL}=P_{PI}=0.8$. In a process with memory trade-offs between higher and lower probabilities between the activities are possible.

E.g. for simplicity's sake let us assume that the activities are statistically independent then the minimum overall probability to hire a suitable candidate is: of $P_{AL}*P_{PI}=0.8*0.8=0.64$. Demanding the same level of overall confidence in a process with memory leads to a trade-off. E.g. the application letter is almost perfect ($P_{AL}=0.9$) and therefore compensates the weaker interview of the candidate ($P_{PI}=0.75$). Although the interview does not meet the confidence level required in the memory-free process the overall confidence in the candidate is higher than in the first case $P_{AL}*P_{PI}=0.9*0.75=0.675$.

Sufficiently and Fully Completed Activities

Along the lines of the discussion on the fuzzy dimension in section *The Fuzzy Dimension* the states in the probability dimension are extended to *closed.completed (P_i)* with P_i a probability between P_M and 1 ($P_M \le P_i \le 1$) where P_M defines the minimum probability that must be met.

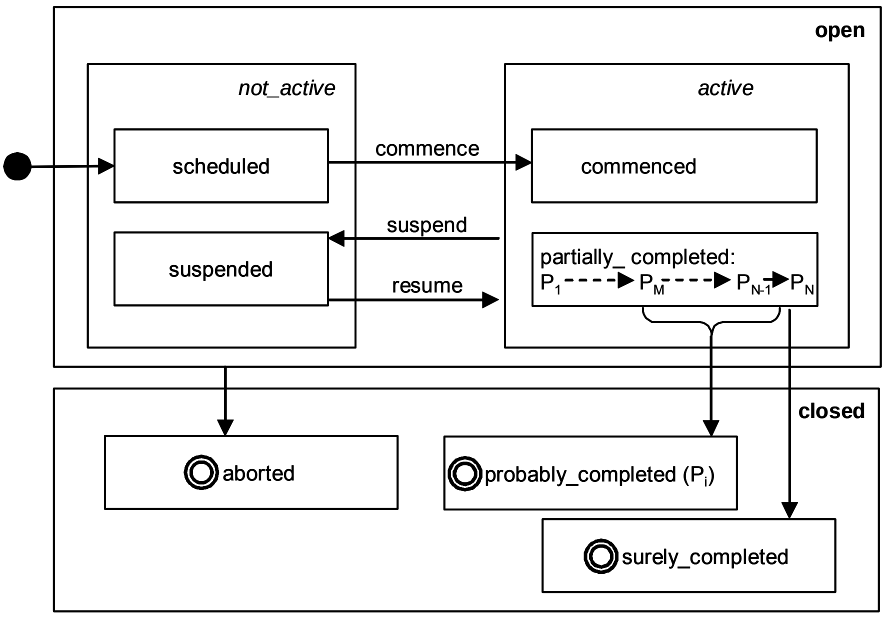

Figure 9: Probability Enhanced Activity Finite State Machine

To clearly distinguish between surely completed activity and activity that are only completed with a certain probability the following abbreviations are introduced:

- state*: closed.completed (P_i)* with $P_M \le P_i <1.0$: *probably_completed (P_i)*
- state: *closed.completed (P_i)* with $P_i=1.0$: *surely_completed*

The nomenclature parallels the one we have already presented for the fuzzy dimension. The corresponding probability enhanced activity finite state machine is show in Figure 9.

Combination of Fuzzy and Probability Dimension

Please note, that there is an important difference between the fuzzy and probability dimension [Zadeh 1995]. The fuzzy dimension measures the *similarity* of an outcome and the post-conditions of an activity (Figure 10). There is no doubt or insecurity as to the degree to which the outcome fulfils the post-conditions of the activity.

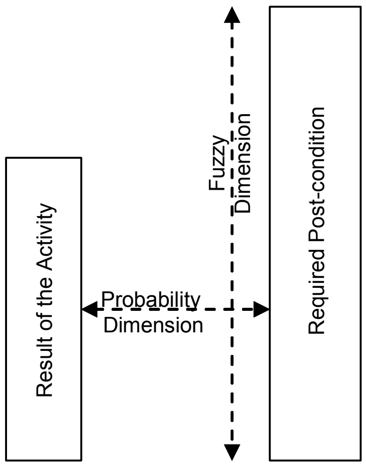

Figure 10: Fuzzy and Probability Dimension

Insecurity is formulated in the probability dimension. This dimension shows the degree of probability that the output of the activity really matches the requirements.

Obviously the fuzzy and probability dimensions are independent from each other. Therefore they can be combined and derived straight forward out of the given finite state machines as shown in the previous sections.

Consequently the combined probabilistic-fuzzy completion states are defined as follows:

- state: *closed.completed (F_i, P_i)* with ($F_M \le F_i$ and $P_M \le P_i$) and (F_i<1.0 and/or P_i<1.0): *sufficiently_completed (F_i, P_i)*
- state: *closed.completed (F_i, P_i)* with F_i=1.0 and P_i=1.0: *fully_completed*

We consider these combined dimensions as principally possible in real-life applications. However a manual customisation of the workflow system which includes the termination of the probabilistic-fuzzy functions and the trade-offs between the activities, would be extremely difficult and complex in real life. Nevertheless we can imagine that automated learning approaches based on artificial intelligence might be suitable for the training of such a system.

CONCLUSION

In this paper we extended the concept of partially completed activities by a process memory and distinguished two independent dimensions, the fuzzy and probability dimension that allow us to describe the reasons for the partial completion of activities in more detail. The process memory allows us to formulate trade-offs between the single activities and make it easier to meet the overall process goal in comparison to a memory-free approach.

Both our extensions lead to an increase in process flexibility in comparison to the approach of Lin and Orlowska. However the downside is that our extensions require more sophistication in workflow design as well as more complex management of the process instances at run-time. Without any automated training method, we think the increased sophistication would be uneconomic for most applications. Therefore the primary focus of future work on this topic will need to be to develop suitable automated learning and customization concepts.

REFERENCES

Carter, B.M.; Lin, J.Y.C; Orlowska, M.E. [2004]: Customizing Internal Activity Behaviour for Flexible Process Enforcement. *Proceed. 15th Australasian Database Conference (ADC 2004)*, Dunedin, New Zealand

DSTC Praxis Project [2004]: Chameleon Website, http://www.dstc.edu.au/Research/Projects/praxis/chameleon/index.html

Dubois, D.; Prade, H. [1982]: A class of fuzzy measures based on triangular norms. *International Journal of General Systems*, 8, 43-61

Leymann, F.; Roller, D. [2000]: *Production Workflow: Concepts and Techniques*. Prentice-Hall, Upper Saddle River, NJ, USA

Lin, J.Y.C.; Orlowska, M.E. [2005]: Partial completion of activity in business process specification. *Proceed. IRMA 2005*, San Diego, CA, USA, 186-189

Marinescu, D.C. [2002]: *Internet Based Workflow Management: Towards a Semantic Web*. John Wiley & Sons, New York

Raposo, A.B.; Coelho, A.L.V.; Magalhães, L.P.; Ricarte, I.L.M. [2001]: Using Fuzzy Petri Nets to Coordinate Collaborative Activities. *Proceed. 9th IFSA (International Fuzzy Systems Association) World Congress and 20th NAFIPS (North American Fuzzy Information Processing Society) International Conference*, Vancouver, Canada, 1494-1499

Reichert, M.; Dadam, P. [1998]: ADEPT*flex* - supporting dynamic changes of workflows without losing control. *Journal of Intelligent Information Systems*, 10, 93-129

Riempp, G. [1998]: *Wide Area Workflow Management*, Springer Verlag, Berlin

Sadiq, S. [2000]: Handling Dynamic Schema Change in Process Models, http://www.dstc.edu.au/Research/Projects/praxis/publications/ssadiq_adc_2000.pdf (accessed 08.07.2005)

Tagg, R.M. [2003]: Dimensions of Flexibility in Workflow Management, *EURO/INFORMS Conference, Istanbul*, 2003

van der Aalst, W.; van Hee, K. [2002]: *Workflow Management - Models, Methods, and Systems*. MIT Press, Cambridge, MA, USA

Werners, B. [1988]: Aggregation models in mathematical programming. In Mitra, G.: *Mathematical Models for Decision Support*. Springer Verlag, Berlin, 295-319

Workflow Management Coalition [1995]: Workflow Management Coalition – The Workflow Reference Model. Document No. WFMC-TC1003

Yager, R.R. [1980]: On a general class of fuzzy connectives. *Fuzzy Sets and Systems*, 4, 235-242

Zadeh, L. [1965]: Fuzzy Sets. Information and Control 8, 338-353

Zadeh, L. [1995]: Probability Theory and Fuzzy Logic are Complementary Rather Than Competitive. *Technometrics*, 37, 271-276

Zimmermann, H.J. [2001]: *Fuzzy sets and its applications* (4th edition). Kluwer Academic Publishers, Boston, MA, USA

Zimmermann, H.J.; Zysno, P. [1980]: *Latent connectives in human decision making*. Fuzzy Sets and Systems, 4, 37-51

COPYRIGHT